Quantifying Bimodality Part 2: A Likelihood Ratio Test for the Comparison of a Unimodal Normal Distribution and a Bimodal Mixture of Two Normal Distributions. Bruno D. Zumbo is

B. W. Frankland
Dalhousie University, Brad.Frankland@dal.ca

Bruno D. Zumbo
University of British Columbia, bruno.zumbo@ubc.ca

Recommended Citation

Quantifying Bimodality Part 2: A Likelihood Ratio Test for the Comparison of a Unimodal Normal Distribution and a Bimodal Mixture of Two Normal Distributions

B. W. Frankland
Dalhousie University

Bruno D. Zumbo
University of British Columbia

Scientists in a variety of fields are often faced with the question of whether a sample is best described as unimodal or bimodal. In an earlier paper (Frankland & Zumbo, 2002), a simple and convenient method for assessing bimodality was described. That method is extended by developing and demonstrating a likelihood ratio test (LRT) for bimodality for the comparison of a unimodal normal distribution and a bimodal mixture of two normal distributions. As in Frankland and Zumbo (2002), the LRT approach is demonstrated using algorithms in SPSS.

Key words: Bimodality, likelihood ratio test, mixture distribution, SPSS.

Introduction

Previously, a method for assessing bimodality using the non-linear algorithms in SPSS was presented (Frankland & Zumbo, 2002). It is a method for modeling complex mixture distributions with a unimodal normal distribution (with 2 free parameters) and with a bimodal mixture of two normal distributions (with 5 free parameters). The current work extends that previous work to the development of a likelihood ratio test (LRT) for bimodality. In this extension, the research question is: Does a bimodal mixture of two normal distributions represent a significantly better fit to the data than a unimodal normal distribution? Here, the fit of the data to the unimodal normal distribution is considered the null hypothesis. The fit of the data to the bimodal mixture of

two normal distributions is considered the alternative hypothesis. The null hypothesis is rejected if it provides a significantly poorer fit to the data.

As noted in Frankland and Zumbo (2002), the techniques developed herein are focused on putative mixtures of normal distributions; they can be applied, in principle, to the comparison of any set of theoretical distributions. Normal distributions were chosen as the focus because it is likely that the normal distribution is a reasonable approximation to the data, either as a single unimodal distribution, or as each component of the mixture of two distributions. It is admitted, a priori, that the solution offered is not an analytical solution to the question of bimodality. The point was to develop an accessible, flexible and, most importantly, accurate method that could be used to test any number of hypotheses. The procedure uses the commercially available statistical package SPSS (most statistical packages should be capable of comparable analyses) to accomplish a Monte Carlo simulation to generate the likelihood ratio distribution for the bimodal/unimodal comparison. Because it can be assumed that most researchers will use this technique to analyze a single (or limited number

B. W. Frankland is an adjunct professor in Psychology at Dalhousie University. Email him at: Brad.Frankland@dal.ca. Bruno D. Zumbo is Professor of Measurement, Evaluation and Research Methodology, as well as a member of the Department of Statistics and the Institute of Applied Mathematics. Email him at bruno.zumbo@ubc.ca.

of) data set, the application is demonstrated within that context.

The Likelihood Ratio Method

A set of empirically determined data, of size *n* (lower case n), is compared to two hypothetical population distributions. It is assumed that the data is represented as a histogram (hereafter, data histogram, or histogram; the term empirical data will refer to the original pre-binned data).

The histogram will define the number of bins, and their statistics (lower limit, center, upper limit) for the subsequent analyses (see Frankland & Zumbo, 2002). This determination should be made in the context of subsequent simulation. The sample size (*n*) is the most important factor for creating bins. The most efficient method is to determine the mean and standard deviation of the sample using a traditional method. These are estimates of the mean and standard deviation (μ, σ) of the corresponding normal population. Thereafter, the number of bins per standard deviation is set to accommodate the expected range and density of scores for any sample of size *n*, from this particular population, $N(\mu, \sigma)$. For example, given *n* = 500, one could use 10 bins per sd, with a full range of z-scores from -5.0 to 5.0. (This point will be discussed more fully later.) These bins can then be adjusted to fit the actual data.

Alternatively, the raw data can be converted to z-scores, and the likelihood ratio test can be conducted using z-scores. The likelihood ratio test is agnostic with respect to the original scale of the data. The use of z-scores is more convenient for testing multiple data sets. However, the fitted statistics for the unimodal and bimodal distributions are not obtained. Here, z-scores were used (see Frankland & Zumbo, 2002 for raw scores).

In the first step, the best-fit parameters for the unimodal and bimodal functions are determined (see Frankland & Zumbo, 2002). The data histogram is first compared to a function that describes a hypothetical unimodal normal distribution (hereafter, unimodal function). With the unimodal function, the free parameters to be determined are the mean (μ) and standard deviation (σ, or variance, σ^2):

$$N(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(X_i - \mu)^2}{2\sigma^2}} \qquad (1)$$

The mean and standard deviation could be easily obtained using traditional methods, but for this application, the mean and standard deviation must be determined using a method that compliments that which is used for the bimodal values.

The histogram is compared to a function that describes a hypothetical bimodal mixture of two normal distributions (i.e., bimodal function). In this case, there is a mean (μ_1, μ_2) and a standard deviation (σ_1, σ_2) for each normal distribution, as well as, the mixture proportion (λ; note that some authors use π, and others use α, for this parameter):

$$B(\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, \lambda) =$$

$$\lambda * \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{\frac{-(X_i - \mu_1)^2}{2\sigma_1^2}} +$$

$$(1 - \lambda) * \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{\frac{-(X_i - \mu_2)^2}{2\sigma_2^2}}$$

$$= \lambda * N_1(\mu_1, \sigma_1^2) + (1 - \lambda) * N_2(\mu_2, \sigma_2^2) \quad (2)$$

Means and variances with subscripts refer to those from the bimodal distribution. In addition to the best-fit parameters, the likelihood that the sample came from a unimodal population, and the likelihood that the sample came from a bimodal population are determined. These two likelihoods are converted to a ratio (the likelihood ratio, LR_{data}).

In the second step, a Monte Carlo simulation is used to create the likelihood ratio test (*LRT*) for bimodality. In this step, a normal distribution is defined, $N(0, 1)$. From this unimodal population, a sample is taken. This sample is converted to a histogram using the same bin parameters as defined previously. The bins used for the raw data must match the bins used for the simulation. The binned sample from the normal distribution is then fitted to a unimodal function and to a bimodal function. Finally, the likelihood ratio for the sample is computed. This process is repeated for a large

number of samples drawn at random from the defined unimodal normal population. From the set of samples, a distribution of the likelihood ratios is created. The likelihood ratio distribution provides a direct assessment of the probability of getting the original likelihood ratio (LR_{data}), if the data were drawn from a unimodal normal population. If that probability is low, then the original data is assumed to be bimodal. This is simply type 1 error rate, which is normally set to $\alpha = 0.05$.

In the simulation there are K samples, each being denoted by k. Each of the K samples is based on n data points, drawn from a normal distribution. Each data point is X_i. These n data points per sample are converted to a histogram: The number and the boundaries of bins are determined by the original data (i.e., bin centers and limits reflect the raw data, X). There are I_i bins ($I = 1$ to I). The initial definition of the bins should encompass the full plausible range of the data (i.e., ideally, the tails should stretch to infinity). For each sample, one likelihood-ratio statistic (LR_k) is produced. The distribution of K likelihood-ratio statistics (LR_k, $k = 1$ to K) statistics provides the test of likelihood ratio of the data (LR_{data}). Note that the original empirical data determines the sample size n. As will be discussed, the sample size is the primary determinate of the number of bins, I. Time, computational resources and desired accuracy determine K. The procedure is demonstrated with a specific example.

The Original Data

For this demonstration, a bimodal data set of $N = 500$ data points was created. The data set consisted of a mixture of two normal distributions. Each distribution was obtained using the SPSS command NORMAL, which generates standard Normal pseudo-random variates. The first distribution was $N(\mu_1, \sigma_1) = N(-1.0, 0.7)$ and the second was $N(\mu_2, \sigma_2) = N(1.0, 1.0)$. Note that the variances are different. The data set consisted of 60% from the first distribution and 40% from the second distribution (sd), and is notated as $B(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda) = B(-1.00, 0.71, 1.00, 1.00, 0.60)$. The raw data had a mean of .169, a standard deviation (sd) of 1.296, a skew of .412 + .109 and a kurtosis of .371 + .218. The median was .329.

These data were converted to z-scores and then binned. By design, there were 10 bins per sd and a full range of $-5.0 \leq z \leq 5.0$. Each bin had a width of .01 sd. There was a single bin centered at $z = 0$ (hence, the bin was defined as $-.05 \leq z \leq .05$). By design, there were 101 bins in total, with the last being $4.95 \leq z \leq 5.05$ and $-5.05 \leq z \leq -4.95$. However, the bins in the tails were widened to encompass the ranges $4.95 \leq z \leq 6.95$ and $-6.95 \leq z \leq -4.95$. This captures the skewness that can manifest in an empirical bimodal distribution (alternatively, one can use a larger range of bins). The resulting distribution is shown in Figure 1.

The z-scores in the raw data ranged from -2.09 to 3.07, and after binning, there were only 50 bins with non-zero counts (see Figure 1). However, the full range of bins must be provided, with zero counts for those that are empty. This is important for the subsequent simulations. There are ways to create empty bins in SPSS, but for a single data set, the manual method is about as fast as any other. The data do not appear to be bimodal, although they are not obviously normal either (it simply seems skewed). Based on counts per bin, the binned distribution, with $I = 50$, produced $(\mu, \sigma)=(-.002, 1.001)$. This is slightly altered from the original raw data. This alteration is important because all subsequent analyses are based on the binned data.

The subsequent analysis uses the bin lower limit (xl), bin center (xc) and upper limit (xu), so the SPSS data file is expected to contain the following variables:

- Binnum: bin number (not actually used, but useful for humans)
- Observed: observed count per bin (X)
- xl: bin lower limit in the original scores
- xc: bin center in the original scores
- xu: bin upper limit in the original scores
- Total: total counts (total number of data points, a constant

Fitting the Original Data

As described previously (Frankland & Zumbo, 2002), when fitting the unimodal or bimodal functions, the algorithm determines the parameters for the unimodal, $N(\mu, \sigma)$, and

Figure 1:The Empirical Bimodal Distribution B(-1.00, 0.71, 1.00, .1.00, 0.60)

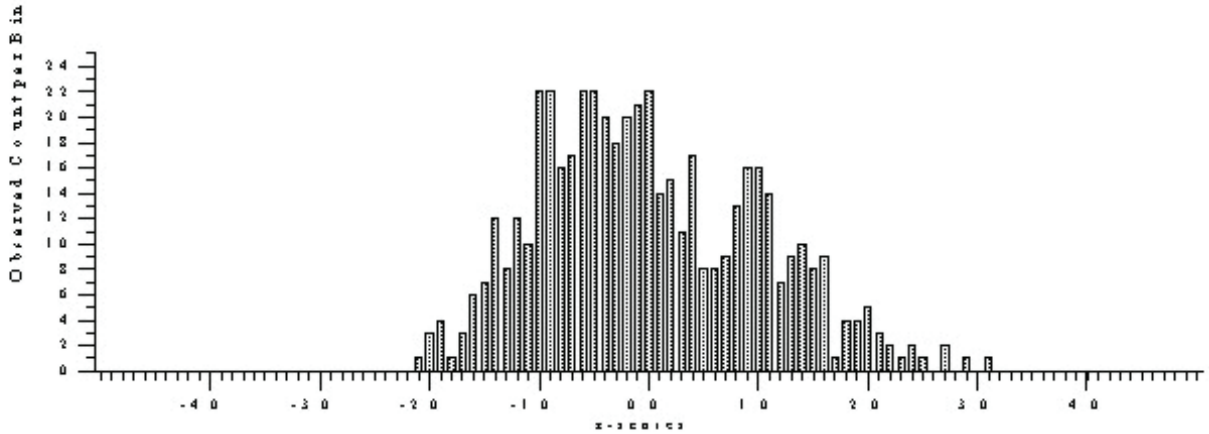Figure 1: The empirical bimodal distribution, B(-1.00,0.71,1.00,.1.00,0.60).

bimodal, $B(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda)$, distributions using an iterative, sequential-quadratic, search algorithm. The algorithm determines the values of μ and σ (or μ_1, σ_1, μ_2, σ_2, and λ) so that the predicted count per bin ($\hat{Y}_i$ or Y'_i) forms the best possible match to the by minimizing the sum of the squared deviations between theoretical count per bin and the actual count per bin (Y_i). Conceptually, the fitting procedure is the same as ordinary, unweighted, least-squares regression (OLS) with X_i being the center of the bin, and Y_i being the actual count per bin. The X_i are transformed non-linearly to create predicted bin count $\hat{Y}_i$ (or Y'_i).

The parameters of the functions are adjusted iteratively until $\hat{Y}_i$ produces the best match to Y_i, ascertained by minimizing the sum of the squared deviations between the predicted and actual, $\Sigma e^2_i = \Sigma(\hat{Y}_i - Y_i)^2$. Relative to OLS, only the method of fitting is different. Note that, in this analysis, every bin has the same contribution to the final solution regardless of the number of scores per bin. The predictions, $\hat{Y}_i$, are not weighted by sample size per bin. This is the simplest approach, but a weighted approach could be developed (i.e., weighted by bin count or, equivalently, bin error).

Because each bin has an equivalent contribution to the final solution, one must choose the bins carefully. After much trial and error, bolstered by post hoc rationalizations, it seems that empty bins for the full range of z-scores should be included in the analysis. This is not a χ^2 solution, though there are links to that methodology. Such data should not be dropped (or trimmed). That is, empty bins between bins with counts, and empty bins representing the tails of the distribution should be retained or added to the histogram. These empty bins in the tails can be combined if necessary.

First, as noted, after much trial and error, the inclusion of empty bins does not seem to make a lot of difference to the final solution. The fitted parameters do change, but the change is within the error of all approaches. However, the inclusion of empty bins has many benefits for the later simulations.

Second, the true functions that are being fitted technically stretch to $\pm\infty$. It is only by virtue of sample size that the data does not stretch to infinity. Having empty bins in the tails forces the functions to go to zero when they should go to zero. Alternatively, the inclusion of empty bins in the tails is equivalent to forcing the regression solution to go through the origin, which is reasonable (the distributions approach zero asymptotically). At this point, the goal is to

find the best fitting parameters to define the populations.

Third, bins in the tails (particularly those distant from the center of the data) represent real, though rare, data. Assuming that the original sampling that led to the data is truly random, deleting such outliers would be equivalent to lobbing off a part of the population. On the other hand, if retained, these outliers have high leverage in the solution. Empty bins in the tails also represent real information (low probability events). The inclusion of empty bins in the tails has the effect of reducing the leverage associated with the retention of outliers. That is, in the solution, a number of bins with $(\hat{Y}, Y) = (\sim 0, 0)$ will balance a few bins with $(\hat{Y}, Y) = (\sim 0, 1)$ or $(\sim 0, 2)$.

Last, bins in the tails can be combined. The fitting functions work by determining the probability of observed data per range of z-scores. The functions use numerical integration with a trapezoid rule. In the tails, the functions are relatively flat (or, at least, approaching linear). Hence, in the tails, the use of a trapezoid rule with wider bins would not introduce large distortions.

Bins should define a reasonable range of data that can incorporate the full range of the data, including the possible range that might occur in the subsequent simulation. A simple definition would use a range of $\pm 5\sigma$ (i.e., z-scores). More sophisticated estimates can be made, particularly with very large samples, but this seems to be a useful default value.

The fitting algorithms for the unimodal and bimodal functions are shown in Listing 1 (also see Frankland & Zumbo, 2002). Note that probabilities are actually computed using a two-trapezoid rule per bin, with three values (X_l, X_c, X_u) per bin (the routine uses proportions per bin, but it could be written to use actual counts).

Listing 1

```
compute          prop = observed/total.
model program   mean= 0.0 sd = 1.0.
compute xa = abs(xl - xc).
compute xb = abs(xu - xc).
compute h1 = (.398942/sd)
          * exp(-(((xl-mean)**2) / (2*sd**2)) ).
```

```
compute h2 = (.398942/sd)
          * exp(-(((xc-mean)**2) / (2*sd**2)) ).
compute h3 = (.398942/sd)
          * exp(-(((xu-mean)**2) / (2*sd**2)) ).
compute  predun = .5 * (h1+h2) * xa
          + .5*(h2+h3)*xb.
cnlr      prop.
          /pred = predun
          /bounds sd gt 0.0001
          /save = predun residun.


model program  mean1 = -1.0 mean2 = 1.0
                sd1 = 1.0 sd2 = 1.0 ratio = 0.5.
compute  xa = abs(xl - xc).
compute  xb = abs(xu - xc).
compute  h1 = (.398942/sd1)
          *exp(-(((xl-mean1)**2)/(2*sd1**2)) ).
compute  h2 = (.398942/sd1)
          *exp(-(((xc-mean1)**2)/(2*sd1**2))).
compute  h3 = (.398942/sd1)
          *exp(-(((xu-mean1)**2)/(2*sd1**2))).
compute  h4 = (.398942/sd2)
          *exp(-(((xl-mean2)**2)/(2*sd2**2)) ).
compute  h5 = (.398942/sd2)
          *exp(-(((xc-mean2)**2)/(2*sd2**2))).
compute  h6 = (.398942/sd2)
          *exp(-(((xu-mean2)**2)/(2*sd2**2))).
compute  predbi = ratio *(.5*(h1+h2)*xa
          + .5*(h2+h3) * xb)
          + (1-ratio)*(.5*(h4+h5)*xa
          + .5*(h5+h6) * xb).


cnlr      prop.
          /pred = predbi
          /bounds sd1 gt 0.0001; sd2 gt 0.0001;
           1.0 ge ratio ge 0.0
          /save = predbi residbi.
```

The constraints (bounds) are placed on the values of variances and the ratio. A constant could be included in the equations. In practice, it seems to make little difference for the fit of either function. More precisely, other factors, particularly the width of the bins, have a greater effect. The routine produces the predicted proportion per bin, $\hat{Y}_i$ (or Y'_i, notated as predun and predbi) and the residual, $e_i = Y'_i - Y_i$ (notated as residun and residbi). These variables are added to the data file. The sum of the residuals should be zero. The sum of the residuals-squared (Σe^2_i) is equivalent to $SS_{Y.X}$ in

OLS (i.e., $SS_{Y.X} = \Sigma e^2_i$), which can be converted to the standard error of estimate, $s_{Y.X}$, and eventually R^2

For the current data, after creating bins that stretched to $\pm 7\sigma$, the analysis using the bimodal function produced $B(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda) = B(-0.489, 0.681, 1.145, 0.497, 0.734)$, with $s^2_{Y.X} = 1.389*10^{-5}$ and $R^2 = .933$. The standard errors on the parameters estimates are 0.046, 0.043, 0.075, 0.064, and 0.041 respectively. When converted back to raw scores using the inverse of the z-transform, these correspond to $B(-0.802, 0.881, 1.312, 0.643, 0.734)$ This compares acceptably with the parameters used to define the population. The analysis using the unimodal function produced $N(\mu, \sigma) = N(-0.163, 1.038)$, with $s^2_{Y.X} = 2.419*10^{-5}$ and $R^2 = .880$. The standard errors on the parameters estimates are 0.044 and 0.036 respectively. Converted to raw scores, one has $N(-0.380, 1.343)$. The fitted functions are layered on top of the original data in Figure 2.

The s^2_Y used for the computation of R^2 is the variance of the counts, not the variance of the original data. The point here is not to compare the parameters returned by algorithm to those of the optimal solution. Rather, the point is to compare the fits using the unimodal and bimodal functions when computed using the same routine. Note that the change in fit is $\Delta R^2 = .933 - .880 = .053$.

For comparison purposes, if the histogram is cut off at the edge of the data (i.e., -2.2 < z < 3.2), but retaining the empty bins between those extremes, one obtains $B(-.486, .686, 1.150, .491, .739)$ with $s^2_{Y.X} = 2.764*10^{-5}$ and $R^2 = .880$. Note that these are within the errors cited above. For the unimodal function, one gets $N(-.176, 1.054)$ with $s^2_{Y.X} = 5.720*10^{-5}$ and $R^2 = .764$.

If all the empty bins are removed (even those between other non-empty bins), one obtains $B(-.486, .686, 1.150, .491, .739)$ with $s^2_{Y.X} = 2.948*10^{-5}$ and $R^2 = .866$, and $N(-.175, 1.055)$ with $s^2_{Y.X} = 4.777*10^{-5}$ and $R^2 = .769$. Clearly, all three methods produce equivalent fits and parameters.

As expected, in all cases the bimodal function produced the better fit between $\hat{Y}_i$ and Y_i. when using the same method (smaller error, higher R^2). It is interesting that $s^2_{Y.X}$ is smaller when there are more bins (i.e., more X and Y points). This is counterintuitive, but it implies that the additional points – the empty bins – have very little error (so that the average error decreases). In addition, note that the choice of bin values does not affect the *relative* fits dramatically. The ratios $s^2_{Y.X,b}/s^2_{Y.X,u}$ are .662, .691 and .710 respectively, while the ΔR^2 are .055, .082 and .098 respectively.

Figure 2: The Unimodal Bimodal Curve Fits to the Empirical Bimodal Distribution
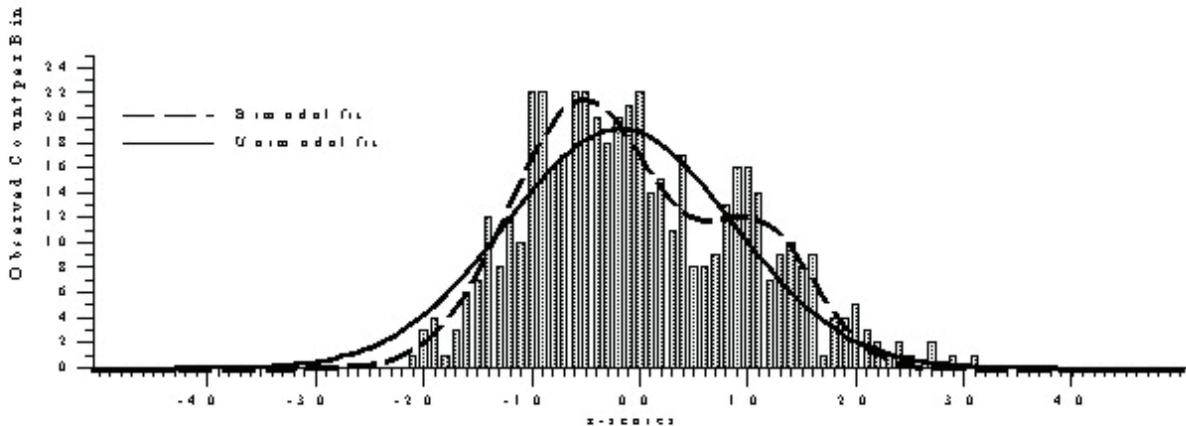

Figure 2: The unimodal and bimodal curve fits to the empirical bimodal distribution.

The Likelihood Ratio for the Original Data

These two fits cannot be directly compared (either R^2 or $s^2_{Y.X}$) because the population with greater number of free parameters will generally produce the better fit. The reason for this is somewhat oblique to the statistical analysis. The premise is that when comparing two theories (i.e., two populations) both theories will have been selected by past research to be reasonable fits to the data (even if only by eye). Hence, both functions will approximately match the data, so the function with the more flexibility (more df's) will generally fit better.

Instead, the likelihood that the data came from a unimodal population can be compared to the likelihood that the data came from a bimodal population (this is almost Bayesian). The probability, or likelihood, of getting the particular set of data if, in fact, that data came from the specified unimodal, $N(\mu, \sigma)$ is $L(N)$. It is also known as L_0, since the simpler unimodal distribution will become the null hypothesis. Similarly, for the bimodal population, $B(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda)$, the probability is $L(B)$ or L_A since the more complex bimodal population will become the alternative hypothesis. In each case, the probability is nothing more than the product of the probabilities for the individual bins, i ($i = 1$ to I).

$$L(N) = L_O = \prod_i P_i(\mu, \sigma) \qquad (3)$$

$$L(B) = L_A = \prod_i P_i(\mu_1, \sigma_1, \mu_2, \sigma_2,) \qquad (4)$$

L_0 and L_A represent the maximum likelihood solutions for each population: They are, in some sense, the best possible fits between the data and the corresponding function, and therefore represent the maximum probabilities (likelihoods) for each hypothesis. The usual mean and standard deviation is, in fact, the maximum-likelihood solution for the normal distribution.

Usually, to compare the two hypotheses, the ratio of likelihoods is computed. This is the likelihood ratio (LR_{data} or Λ_{data}) for the data. In the likelihood ratio, for reasons that will be obvious momentarily, the simpler or null hypothesis is placed in the numerator.

$$LR_{data} = \Lambda_{data} = {L_O}/{L_A} \qquad (5)$$

This ratio will be bounded by (0.0, 1.0). A ratio near 0 indicates that the alternative hypothesis (the bimodal distribution) is a much better fit, and a ratio near 1 indicates that both hypotheses provide equivalent fits. A ratio much greater than 1 should be impossible since the unimodal should not be able to provide a better fit than the bimodal distribution.

Because there are many computational advantages, one usually works with the natural logarithm of the likelihood ratio. Hence, one usually has:

$$-2\ln(\Lambda_{data}) = -2\ln\left({L_O}/{L_A}\right) \qquad (6)$$

$$= 2\ln(L_A) - 2\ln(L_O)$$

$$= 2\ln[\prod_i P_i(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda)] - 2\ln[\prod_i P_i(\mu, \sigma)]$$

$$= 2\sum_i \ln[P_i(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda)] - 2\sum_i \ln[P_i(\mu, \sigma)]$$

$$= 2\sum_i \{\ln[P_i(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda)] - \ln[P_i(\mu, \sigma)]\}$$

$$= LLR_{data}$$

The value of logarithm of the likelihood ratio, $LLR = -2\ln(\Lambda)$, is bounded by $(-\infty, \infty)$, although very large positive or negative values ($<-1000, >1000$) would not be expected. A zero indicates equivalent fits, negative values imply that the unimodal is a better fit while positive values imply that the binomial is a better fit. Large positive values lead to rejection of the null hypothesis. The important point (for algorithms) is that by using $\ln(L_A)$ and $\ln(L_0)$, one converts the previous products and their ratio into a series of sums. Most importantly, the difference between the two hypotheses can be computed on a bin-by-bin basis, and then summed.

To find the ratio, the likelihood that the data comes from the best-fit unimodal distribution must be determined, along with the likelihood that the data comes from best-fit bimodal distribution. This has not been detailed

in previous work (i.e., Frankland & Zumbo, 2001). To obtain these probabilities, for each bin, the probability of an observed count, given the theoretical bin count must be determined. If it is assumed that, within any single bin, i ($i = 1$ to I), the counts per bin follow a Poisson distribution (a normal distribution per bin may also be assumed), the probability of any observed count (Y_i) can be obtained given the theoretically predicted count ($\hat{Y}_i$ or Y'_i). Within each bin, the probability for any particular count is (subscripts have been dropped for clarity):

$$p(Y/\hat{Y}) = \frac{\hat{Y}^Y}{Y!}$$

$$\ln[p(Y/\hat{Y})] = Y \ln(\hat{Y}) - \ln(Y!) - \hat{Y} \quad (7)$$

Essentially, the predicted count, $\hat{Y}$, is nothing more than a non-linear transformation of the bin value (e.g., collectively, xl, xc, xu). Listing 2 provides a method for computing the probabilities in each bin, given predicted counts from the unimodal and bimodal functions. Note that since the Poisson distribution uses counts, not proportions, proportions (of Listing 1) are converted into counts. The loop simply computes the factorial. The loop should not be executed if the observed count is zero.

Listing 2: Poisson Probabilities per Bin: Unimodal and Bimodal Functions

```
compute  expectun = predun * total.
compute  expectbi = predbi * total.
compute  poisun = (expectun**observed)
           * exp(-1*expectun).
compute  poisbi = (expectbi**observed)
           * exp(-1*expectbi).
loop       #i = 1 to observed.
 compute   poisun = poisun / #i.
 compute   poisbi = poisbi / #i.
end loop.
compute    lnpoisun = ln(poisun).
compute    lnpoisbi = ln(poisbi).
compute    llrdata = 2 * (lnpoisbi - lnpoisun).
```

Bin-by-bin, the probabilities (poisun and poisbi) are converted to logs and then subtracted. In the final step, the differences would be summed to create the ratio LLR_{data}.

However, the use of logarithms has many benefits. As noted in Equations 6 and 7 (and Listing 2), the factorial depends on the observed count and as such, is the same for both the unimodal and bimodal functions on a bin-by-bin basis. When converted to logs, the factorials become sums that cancel in each bin. Hence, the pesky loop to compute the factorial is not needed, removing complications arising from bins with zero counts. The early transition to logarithms also prevents possible overflow errors in the event that there is a large difference between the observed and predicted, and underflow errors in the event that $e^{-\hat{Y}}$ is very small. Hence, Listing 2 is revised as follows:

Listing 2 Revised: LRT using Poisson Probabilities per Bin

```
compute  expectun = predun * total.
compute  expectbi = predbi * total.
compute  lnpoisun = observed
           * ln(expectun) - expectun.
compute  lnpoisbi = observed
           * ln(expectbi) - expectbi.
compute  llrdata = 2 * (lnpoisbi - lnpoisun).
```

Comment Sum the ln(LRdata) using the simple frequencies command.

```
frequencies prop expectun residun lnpoisun
           expectbi residbi lnpoisbi llrdata
           /format = notable
           /statistics = mean stddev variance
           minimum maximum sum.
```

The sum of llrdata is $LLR_{data} = -2\ln(\Lambda_{data})$, easily obtained from the descriptives or frequencies command of SPSS

For the current data, when using bins in the full range of $-5.0 < z < 5$, the $LLR = 28.645$. Note that this is far from the value of zero that would be expected if the data were truly unimodal. However, this is not surprising given that the data was designed as bimodal.

With the more restrictive range $-2.2 < z < 3.2$ (i.e., cut the histogram at the edge of the data, but retaining intervening empty bins), the value is $LLR = 4.259$ (but the simulation distribution changes accordingly). When no

empty bins are included the value is $LLR = 2.363$. One then uses these values to determine whether the sample is more likely to have come from a unimodal or bimodal population.

The other variables in the frequencies command provide quick, but useful, checks on the analysis. The sums of the proportion per bin (prop) or the predicted per bin (predun, predbi) should be one. The sums of the observed, expectun and expectbi should equal the number of data points ($N = 500$). The residuals should sum to zero. In addition, the variance of the residuals is essentially the squared-standard error of estimate ($s^2_{Y.X}$) for each function. The output includes the variance of the bin counts. This is s^2_Y. From this, one can compute the correlation $R^2 \approx 1 - s^2_{Y.X}/s^2_Y$. Although the CNLR routine will provide R^2 directly, this computation is useful when computing R^2 in a simulation (herein it serves as a further check).

The Sampling Distribution of Log-Likelihood Ratio (LLR)

The last step is to decide whether or not the observed ratio, $LLR_{data} = -2\ln(\Lambda_{data})$, is reasonable if, in fact, the null hypothesis is true. This is the likelihood ratio test (LRT) or more properly the log- likelihood ratio test ($LLRT$). To make this decision, one needs the theoretical sampling distribution of $LLR = -2\ln(\Lambda)$. This theoretical distribution is focused on the possible values of LLR when the data is taken from the defined unimodal normal distribution (i.e., from $N(\mu, \sigma)$, or L_0). The empirical or theoretical bimodal population is irrelevant to the creation of this distribution.

To create this distribution, one defines a normal distribution, and then takes a sample (notated by k) from that distribution. That sample is fit with a unimodal function, with a bimodal function and then the LLR_k is determined. If the two functions provide equivalent fits, the LLR_k is expected to be near 0.0, but in fact, a value slightly greater than 0.0 is expected (if the data is unimodal, the bimodal function will provide a better fit given its greater flexibility). This sample has the same sample size as the original data (n; called total in Listing 2).

The process is repeated for K samples to create the sampling distribution of LLR. The

mean of this distribution is expected to be slightly greater than 0.0. The value of K reflects the desired precision in the final likelihood ratio distribution, weighted by the amount of patience.

If the observed value of $LLR_{data} = -2\ln(\Lambda_{data})$ for the single sample under consideration is unlikely given that distribution, then the null hypothesis is rejected – it is then concluded that the data is bimodal. That is, the null hypothesis is that the data is unimodal. In that case, the LLR_{data} should be near 1.0 (the mean of the likelihood ratio distribution). If the data is actually bimodal, then the value of LLR_{data} will be unexpectedly large. The usual criteria regarding Type 1 Error Rate (α) can be employed as the basis for the decision. If the LLR_{data} is one of those values that is so large that it would only be expected to occur 5% of the time (if the null hypothesis, L_0, were to be true), then it is unlikely and the associated L_0 is unlikely, and the null hypothesis is rejected.

A Monte-Carlo simulation is used to create the sampling distribution of LLR. There are many nuances that can be varied for the simulation, but the point here is to create a basic template that can be used broadly or adapted to specific situations.

For that simulation, there are a couple of important observations. Firstly, for the simulation, the scaling of the bin centers (or limits) is irrelevant to the issue of computing the sampling distribution of the LLR. The routine assesses the relative match of the Y_i to the predictions based on the best fitting unimodal function ($\hat{Y}_{i,0}$) and bimodal function ($\hat{Y}_{i,A}$). Both the unimodal and bimodal functions use the same bin centers. Both $\hat{Y}_{i,0}$ and $\hat{Y}_{i,A}$ are simply non-linear transforms of the same underlying bin centers. Hence, the data can be conveniently rescaled so that the bins are coded in terms of z-scores, with a certain number of bins per standard deviation. The data only needs to be coded in the original units for discussion of the actual unimodal or bimodal means, variances, as well as the λ.

Secondly, the routine must run unattended. This requires careful consideration of the bin definitions. When taking random samples from a population (unimodal or bimodal), every sample in the simulation will

produce different data. Each data point will be assigned to one bin in the histogram. Therefore, one must provide the full range of possible bins for the data to fall into. Most critically, there cannot be any missing bins within the range of the data. That is, each sample may not produce a non-zero count for every bin, but for the simulation as a whole, every individual data point must fall within some bin. Hence, all possible bins must be defined a priori. However, this is not difficult to do because one knows that the K samples are derived from a normal distribution with known mean and variance.

It is important to remember that sampling from the theoretical normal distribution may produce data that extends beyond the range of the original raw data. Hence, the creation of the likelihood ratio distribution should allow for bins that encompass far more range than that of the original data. The bins should extend as far as is reasonable given the theoretical normal population and the empirical data to be tested. It is also appropriate (or safe) to retain a wide bin for each tail to capture the occasional data point that goes beyond the expected range.

The bins in the simulation must match those used with the histogram for the original data: If not, the wrong sampling distribution is created. Note that the LLR_{data} previously computed depended on the types of bins used. Hence, the bins used to construct the histogram for the original data, and the bins used for the simulation must be the same. This is most easily accomplished using a fixed number of bins per sd (e.g., 5 or 10 depending on n), with a range of bins that is adequate for both the raw data and the simulation. Bins in the tails can be made wider without affecting the solution. This is the logic behind the aforementioned range of -5.0 < z < 5.0 with bin for the lower tail expanded to -5.0 to -7.0 and the bin for the upper tail expanded to 5.0 to 7.0. For example, if one is working with original data that contains $n = 100$ data points, the range of z-scores should be about +3.5 standard deviations with 4 or 5 bins per sd (hence 28 to 35 bins in total). This should result in reasonable counts near the center of the distribution, while allowing for the increased spread that is characteristic of a bimodal distribution.

If an original sample of $n = 100$ should have a large proportion of data extending beyond 3.5 standard deviations, one should question the need for a test based on the null hypothesis of a unimodal normal distribution (i.e., the data is clearly not normal). That is, only 0.047% of a normal distribution is beyond 3.5σ, which, for a sample of 100, is no scores. For 500 data points, the range should be expanded to at least +4 bins (0.0063% of a normal distribution) or +5 (0.00057% of a normal distribution). Of course, wider limits are needed because nothing is truly normal, and one must have sufficient range to encompass the original histogram which is not likely normal.

Creating the Sampling Distribution

For the Monte-Carlo simulation, the only significant addition to the previously cited routines is the automated data generation. In the following, it is acknowledged that many of the routines can be simplified or streamlined. This presentation was chosen to maintain the clarity of the logic.

To find the distribution of the LLR, the Monte-Carlo simulation (MCS) uses the full range of bins defined by the data and simulation. These bins are notated by z-scores since this is convenient. The MCS then takes K samples, each of size n, from this population. This results in one large data file. That file contains individual data points. That large data file is split and each sample is analyzed separately (and automatically). Each sample is converted to a histogram using the aforementioned bin sizes. Again, the bin sizes for the theoretical distribution are perfectly matched to those used with the real data, and the real data must have defined and used a sufficient range of bins for the entire simulation. Then, for each sample, the LLR is computed. Finally, all samples are reduced to a single data file containing the distribution of LLR_k. This distribution can be plotted, or more simply the necessary critical values can be obtained.

The first part of the process is shown in Listing 3. This generates K samples of size n. There are a couple of tricks to be discussed momentarily. Note the random seed.

Listing 3: Generating K samples of Size n

```
set seed        random.
input program.
compute      #mean = 0.
compute      #std = 1.
loop         #K = 1 to 1000.
+ loop       #N = 1 to 601.
+ compute    K = #K.
+ compute    meanbin = 51.
+ compute    binpersd = 10.
+ compute    total = 500.
+ do if ( #N le 101 ).
+ compute    N = -1.
+ compute    zscore = 0.
+ compute    binnum = #N.
+ compute    xl = (binnum-meanbin - .5)
                / binpersd.
+ compute    xc = (binnum-meanbin)
                / binpersd.
+ compute    xu = (binnum-meanbin + .5)
                / binpersd.
+ end if.
+ do if ( #N gt 101 ).
+ compute    N = #N - 101.
+ compute    zscore = normal(#std) + #mean.
+ compute    binnum = rnd(zscore * binpersd)
                + meanbin.
+ end if.
+ end case.
+ end loop.
end loop.
end file.
end input program.
execute.

frequencies  binnum.
if ( binnum le 1 )      binnum = 1.
if ( binnum ge 101 )  binnum = 101.
if ( binnum eq 1 )      xl = -6.95.
if ( binnum eq 101 )  xu = 6.95.
if ( binnum eq 1 )      xc = (xu - xl) / 2 + xl.
if ( binnum eq 101 )  xc = (xu - xl) / 2 + xl.
execute.
```

First, to generate *n* data points, *n*+101 data points are generated. The extra 101 data points are a trick. They are place holders to ensure that every data set has the same range of bins. They define the bin sizes (in z-scores). The 101 comes from the desire to have a range of -5.0 < z < 5.0, with 10 bins per standard

deviation. Note the variable meanbin and binpersd. There is one odd bin at the center. This can be altered to suit the circumstances (i.e., a different number of bins per standard deviation; a range of z-score range of bins).

Second, bins are actually numbered from 1 to 101, rather than from -50 to 50. The variable meanbin defines the center bin. The values of xl, xc, and xu define the limits (lower, center, upper) of the bin in terms of z-scores. These are most useful for verifying the execution of the program. The frequencies command simply serves to check if any data exceeded the expected range of z-scores. Note that the tails are artificially widened after the data is created.

This routine creates a data file that contains the following variables per case:

K	sample number
meanbin	the center bin
binpersd	the number of bins per standard deviation
xl	the lower (left) limit of the bin, in z-scores
xc	the center of the bin, in z-scores
xu	the upper (right) limit of the bin, in z-scores
total	the total number of data points per sample
N	datum number (not actually used, but useful for humans)
	N = -1 indicates a bin place holder
zscore	the z-score of the created datum
binnum	the conversion of the zscore to a bin number

Note that some of the defined values are constants for all cases (for each data point). This is essentially the same as in original data.

The processing continues in Listing 4. This large data file is split into K smaller files for individual analyses. The SPSS SPLIT FILE function accomplishes this. The data is then sorted (within each sample is faster) by bin number, and collapsed by bin number using the AGGREGATE function. This creates a histogram, for each sample, by counting the number of times each binnum was presented in the data (the line observed = *n*(binnum)). Other variables are collapsed as well. Note that meanbin, binpersd, total, xl, xc, and xu are all constants. Hence, taking the first occurrence

(within each sample) is the most efficient manner to get these values: It does not require any computations by SPSS. Also note that the breaking variables (*K* and binnum) are automatically included in each sample, while the variable zscore is dropped (one could take the mean of zscore to obtain the true bin center).

Listing 4: Converting Data to Histograms, then Cleaning

```
split file by K.
sort cases by k binnum.
aggregate     outfile = *
              /break = K binnum
              /meanbin binpersd total =
               first(meanbin, binpersd, total)
              /xl xc xu = first(xl, xc, xu)
              /observed = n(binnum).
execute.
compute       observed = observed - 1.
frequencies   observed.
if ( observed lt 0 )  observed = 0.
```

In addition, recall that the first 101 values of data only served to ensure that every bin existed (i.e., they were place holders used to define bins). This case would have been included in the count of values per bin number (binnum). Hence, every bin has one count (i.e., observed) too many, so one must subtract one from every value of observed. Note that if the range of bins was not defined sufficiently (the initial 101 bins), there will be a negative count in some bins. This would create havoc with the routines, so a check is used to force the count per bin (observed) to be greater than or equal to zero. The frequencies command is a better check. In fact, if there are negative bin counts (after subtracting one), the analysis should be re-run, or widen the tails still further. Technically, this would also require recomputing LLR_{data} because the bins used for the simulation must match the bins used for the data.

Listing 5 provides the fitting of the two functions and the computation of LLR_k. It is essentially a repeat of previous discussions (particularly Listing 1). Note the set results none command. This turns off the outputting of results which is very useful in a simulation. In addition, split file processing is still engaged.

This is the slowest part of the routine (get a large coffee).

Listing 5: Fitting Each Sample with the Bimodal and Unimodal Functions to Obtain LLR

```
set results none.

compute  prop = observed / total.

model program   mean=0.0 sd = 1.0.
compute  xa = abs(xl - xc).
compute  xb = abs(xu - xc).
compute  h1 = (.39894228/ sd)
             *exp(-(((xl-mean)**2) /(2*sd**2)) ).
compute  h2 = (.39894228/ sd)
             *exp(-(((xc-mean)**2) /(2*sd**2)) ).
compute  h3 = (.39894228/ sd)
             *exp(-(((xu-mean)**2) /(2*sd**2)) ).
compute   predun = .5*(h1+h2)*xa
             + .5*(h2+h3)*xb.
cnlr      prop
          /pred = predun
          /bounds sd gt 0.0001
          /save = predun residun
          /criteria iter 100.

model program  mean1=-1.0 mean2=1.0
               sd1=1.0 sd2=1.0 ratio=0.5.
compute  xa = abs(xl - xc).
compute  xb = abs(xu - xc).
compute  h1 = (.39894228/ sd1)
         *exp(-(((xl-mean1)**2)/(2*sd1**2))).
compute  h2 = (.39894228/ sd1)
         *exp(-(((xc-mean1)**2)/(2*sd1**2))).
compute  h3 = (.39894228/ sd1)
         *exp(-(((xu-mean1)**2)/(2*sd1**2))).
compute  h4 = (.39894228/ sd2)
         *exp(-(((xl-mean2)**2)/(2*sd2**2))).
compute  h5 = (.39894228/ sd2)
         *exp(-(((xc-mean2)**2)/(2*sd2**2))).
compute  h6 = (.39894228/ sd2)
         *exp(-(((xu-mean2)**2)/(2*sd2**2))).
compute  predbi = ratio *(.5*(h1+h2)*xa
          + .5*(h2+h3)*xb)
          + (1-ratio)*(.5*(h4+h5)*xa
          + .5*(h5+h6)*xb) .
cnlr      prop
          /pred = predbi
          /bounds sd1 gt 0.00001;
           sd2 gt 0.00001;
```

```
    1.0 ge ratio ge 0.0
    /save = predbi residbi
    /criteria iter 100.

compute  expectun = predun * total.
compute  expectbi = predbi * total.
compute  lnpoisun = observed * ln(expectun)
            - expectun.
compute  lnpoisbi = observed * ln(expectbi)
            - expectbi.
compute  llrdata = 2*(lnpoisbi - lnpoisun).
execute.
```

Finally, as shown in Listing 6, the data are collapsed once again (using the AGGREGATE function) to create one case (i.e., one line in the data file) per sample. This one case contains all the essential information for the entire sample. The most important is the LLR_k from which the sampling distribution of LLR can be created. The use of percentiles in the FREQUENCIES command provides the standard critical points directly, but the distribution can also be created.

Listing 6: The Sampling Distribution of LLR

```
aggregate outfile = *
        /break k
        /nbins = n(total)
        /count = sum(observed)
        /sumy predun residun predbi residbi=
         sum(prop, predun, residun, predbi,
         residbi)

        /sdy sdresun sdresbi =
          sd(prop, residun, residbi)
        /llr = sum(llrdata).

compute   R2bi = 1 - (sdresbi**2 / sdy**2).
compute   R2un = 1 - (sdresun**2 / sdy**2).
compute   chgR2 = (sdresun**2 - sdresbi**2)
          / sdy**2.

frequencies  variables = llr chgr2
        /percentiles = 90 95 99
        /statistics = mean stddev variance
         minimum maximum median
         skewness seskew kurtosis sekurt
        /order= analysis.
```

For the current data, using the bin sizes of the original data, with $n = 500$ and $K = 1,000$, one obtains the following distribution of LLR (see Figure 3).

From this, it can be determined that 5% of the distribution for LLR exceeded the critical value of .186, so the observed value of $LLR_{data} = 28.645$ is significant. The hypothesis that the data came from a unimodal distribution is rejected, using a type 1 error rate of $\alpha = .05$. This is not surprising given the population definition $B(-1.0, 0.7, 1.0, 1.0, .6)$, the large sample size ($N=500$), range of bins $-5.0 < z < 5.0$ and the 10 bins per standard deviation. Note that it is the sample size that allows for a large range of z, with a small z per bin. The 10% point was .153, and the 1% point was .226. The mean for the distribution was .002 and the standard deviation was .122 (skew: $-.573 + .077$; kurtosis: $.413 + .155$). Note that the mean is quite close to the expected value of zero.

The CNLR function does not allow the correlation (R^2) to be saved per sample. However, R^2 can be computed per sample from $R^2 \approx 1 - s^2_{Y.X}/s^2_Y$. This can also be converted to a distribution. Given the unimodal and bimodal R^2 per sample, one can create ΔR^2, and create the distribution of ΔR^2. The sample ΔR^2 can also be compared to this distribution, or this empirically determined sampling distribution of ΔR^2 can be compared to the theoretical distribution of ΔR^2 with $df_1=3$ and $df_2=n-5$.

For the current data, the change in fit was $\Delta R^2 = .053$. For the distribution of ΔR^2, the critical points were .0000916 at 10%, .000115 at 5% and .000152 at 1%. The mean was .00000153 and the standard deviation .00000064. Given that the observed ΔR^2 was .053, the hypothesis that the data came from a unimodal distribution is rejected.

Te standard deviation function in the AGGREGATE command (e.g., sdresun = sd(residun)) returns the inferential form of the standard deviation which in these simulations is $\Sigma e^2_i / (I-1)$ (where I = number of bins). However, the CNLR algorithm provides the standard error of regression ($s^2_{Y.X}$), and this is used to compute R^2 for each sample. Thus, the $s^2_{Y.X}$ cited in the output of the unimodal modal is $\Sigma e^2_i / (I-2)$, and the $s^2_{Y.X}$ cited for the bimodal modal is $\Sigma e^2_i / (I-5)$. Therefore, technically, the R^2 cited in the

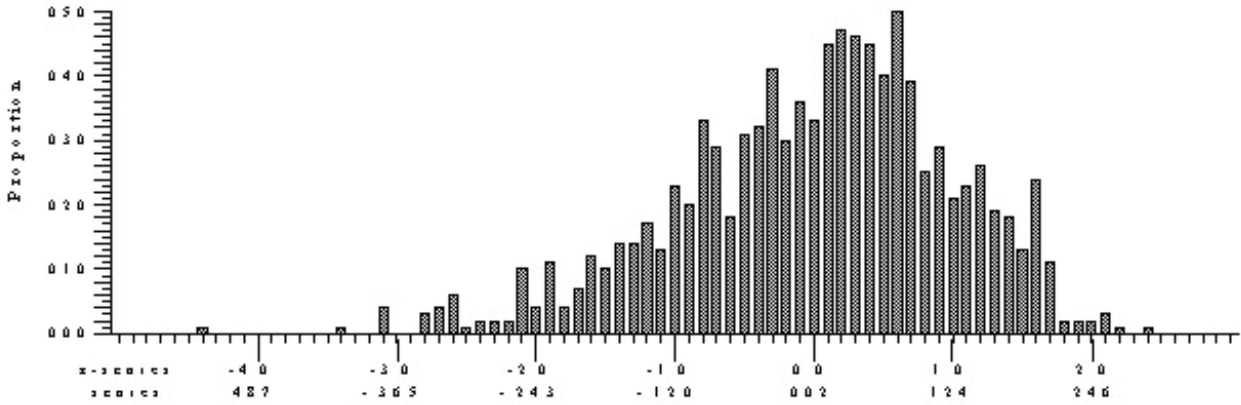Figure 3: The Likelihood Ratio Distribution ($K = 1,000$, $N = 500$)

Figure 3: The Likelihood Ratio Distribution (K - 1000, N - 300).

output of the CNLR for the single sample cannot be directly compared to the distribution of R^2 determined from $R^2 \approx 1 - s^2_{Y.X}/s^2_Y$. However, the difference is slight, and this entire process only estimates the distributions (i.e, it is not an analytic solution). A correction could be applied if desired (I-1 / I-df$_{model}$), which would be useful if generating very large simulations.

Extensions

The CNLR (or NLR) command, with the use of the subcommand /outfile= aaaa.bbb, allows various parameters from each sample to be saved for future analyses. For example, the fitted parameters can be obtained per sample (μ, σ) and (μ_1, σ_1, μ_2, σ_2, λ) so to map the parameter space. When examining these values, it should be kept in mind that the algorithm will occasionally flip the order of μ_1 and μ_2, so, before computing any interesting statistics, one should insure that μ_1 is less than μ_2 (flipping μ_1 and μ_2 also requires flipping σ_1 and σ_2, as well as inverting λ). The output file also contains the SS$_{Y.X}$, the number of cases and the split file number.

A χ^2 test of the fit can also be obtained, by computing ($\hat{Y}_i$ - Y_i)^2/$\hat{Y}_i$, per bin before collapsing the data. This is not advocated because the sum can create overflow errors. The reduced χ^2 can also be used. It is interesting to

note that in the special case when the null is a normal distribution, and the alternative is a mixture distribution of two normal distributions with equal variances ($\sigma_1 = \sigma_2$), the sampling distribution of LLR is known to be related to the χ^2 distribution:

$$\chi^2_\upsilon = -2\ln(\Lambda)$$

$$= -2\ln\left(\frac{L_O}{L_A}\right)$$

$$= 2\ln(L_A) - 2\ln(L_O) \qquad (8)$$

The df (υ) for the χ^2 distribution is equal to the difference in the number of parameters fitted. In this special case, $\upsilon = 2$: There are two parameters for the unimodal normal distribution, $N(\mu,\sigma)$, and four for the bimodal mixture of two normal distributions $B(\mu_1, \sigma, \mu_2, \sigma, \lambda)$,. Hence, in that special case, $LLR_{data} = -2\ln(\Lambda_{data})$ can be compared to the χ^2_2 distribution (see McLachlan, 1987, for a more extensive discussion), though this equivalency assumes that the computation of the expected frequency per bin follows a Normal, rather than Poisson distribution. If the LLR_{data} exceeds the critical value for $\chi^2(2)$, then the null can be rejected. The χ^2_2 distribution can also be compared to the LLR distribution obtained herein. However, these constraints are

not acceptable in the general case. The current procedure allows the use of unequal variances, and the current procedure can be extended to any non-normal distributions.

Because the scaling of the bin centers (or limits) is irrelevant (i.e., X_i), z-scores can be used for any data set. When the data set is converted to a histogram, the important issues are the number of bins per standard deviation and the full range of bins. The actual scaling of the bins is irrelevant (to the computation of LLR_{data}). As such, any particular data set can be converted to a standard histogram, with a set number of bins and range of bins. The LLR_{data} can be determined for that standard histogram. This LLR_{data} could then be compared to tabulated values of critical LLRs for particular values type 1 error rate (α). That is, using SPSS, tables of critical values can be created for various combinations of total sample size, bin size (bins per sd), and range of bins. This would avoid all the tedium of running this simulation for every data set. The simulation could be saved for non-tabulated situations. Arguably, it is still better to complete the entire simulation so that the bins can be carefully tailored to data. Note that the number of samples in the simulation (i.e, K) should not be an issue. That is, K reflects desired precision and reliability. Every simulation should produce approximately the same critical values (always remember that the fitting process is iterative, not algorithmic).

Power

The previous simulations can be used to compute power for any given exact alternative to the null. The exact alternative would specify the parameters in $B(\mu_1, \sigma_1, \mu_2, \sigma_2, \lambda)$, and then use this as the population for the simulation, in place of $N(\mu, \sigma)$. The proportion of sample LLR's that exceeded the previously defined critical values for the null would then be determined. For this to work, the bimodal population must use the same range of bins and the same bin size as the corresponding null hypothesis.

Often, when computing power, a wider range of bins is needed because data in the tails are more common from a bimodal population. This implies that the LLR test for the unimodal population would need to be computed with a larger range of bins.

In addition, when the bimodal population is not symmetric (i.e., $\sigma_1 = \sigma_2$, and $\lambda = .5$), it is more difficult to get the population centered in the histogram. The middle of the histogram should correspond to the mean of the bimodal distribution, so that there is sufficient range in the tails. This is a pragmatic issue since the bimodal population, and hence every random sample from it, is fit with the unimodal and bimodal function on the basis of the same histogram.

Conclusion

This work has been a demonstration of the application of commonly available statistical software, in this case SPSS, to solving the problem of assessing putative mixture distributions, particularly decisions comparing a unimodal normal distribution to a bimodal mixture of two normal distributions. Routines were developed to enable anyone to determine the best-fit statistics for fitting data to a unimodal normal distribution or a bimodal mixture of two normal distribution, to then use those parameters to generate the LLR, and finally, to generate the sampling distribution of the LLR.

These routines have been developed and refined over a number of versions of SPSS from 6.0. to 11.5. In fact, the routines were initially developed within SPSS 4.0, running under VMS 8.0, on a VAX 4500. Different version might require minor modifications. In addition, routines have be developed and run on a variety of hardware. On a 1,000 MHZ Duron with 1.256 Gigs of memory, a simulation with $K = 1,000$, and $N = 500$ required about 15 minutes. A 600 MHZ, Pentium 4 with 256 Megs of memory increased this to about 15 minutes. By contrast, similar simulations on a 40 MHZ AMD 386 had to be run overnight. Interestingly, the VAX also required an overnight batch job.

When setting up, the process is simple and relatively efficient: simply convert the empirical data to z-scores and then create a histogram with an appropriate number of bins per sd and an appropriate range of z-scores. This depends primarily on the sample size. The simulation to create the LLR distribution uses the same bin size and range. The variable bin

widths could be used, with narrower bins near the center of the distribution and wider bins in the tails. As long as the bins form a mutually exclusive and exhaustive set for the range of interest, this is not a problem. In fact, it might be more optimal in the long run to develop algorithms that use bins that represent constant probabilities under the normal distribution.

The method can be adapted to non-normal distributions or to mixtures of non-normal distributions. There is unlimited flexibility in the choice of fitting functions. The process creates an empirical sampling distribution for whatever hypotheses are being tested.

As noted predicted bin counts could be generated using other methods, in particular the normal distribution. That route was not presented here because it the use of normal distribution to predict bin counts resulted in a test with lower power. However, that method is more closely tied to the χ^2 test of fit, and the LLRT approximation to the χ^2.

The second advantage is that the algorithm can be modified to obtain greater accuracy. Non-linear regression using a least-squares error term assumes that the theoretical error is a constant for all values of the independent variable. That is, every bin, regardless of its count, has the same contribution to the final solution. However, the error of a count (if Poisson statistics are valid) is the square root of the count. Hence, relative errors per bin increase as the count decreases. This can be used as a control in the CNLR routine. SPSS non-linear regression allows one to specify the error term. Hence, a weighted least-squares (non-linear) regression approach could be used.

In summary, the routine works; however, it must be cautioned that this algorithm is only considered an interim solution to the problem – one of many (cf., Eriksen & Eriksen, 1972; Eriksen & Yeh, 1985; Hartigan, 1974; Jones & McLachlen, 1990; Müller & Sawitzki, 1991; Roeder, 1990, 1994; Yantis, Meyer and Smith, 1991; Yellott, 1971).

Hopefully, a proper fully parametric method for assessing bimodality will be developed, one that extracts all the information contained within each individual data point rather than working through the intermediary of a histogram. However, even if a proper parametric method is developed, it will necessarily be tied to particular parent distributions. As such, the algorithms developed herein will continue to serve some purpose with other non-normal parent distributions.

References

Bevington, P. R., & Robinson, D. K. (1992). *Data Reduction and Error Analysis for the Physical Sciences*. Toronto:McGraw Hill, Inc.

Do, K., & McLachlan, G. L. (1984). Estimation of mixing proportions: A case study. *Applied Statistics*, *33*(*2*), 134-140.

Eriksen, C. W., & Yeh, Y. Y. (1985). Allocation of attention in the visual field. *Journal of Experimental Psychology: Human Perception and Performance*, *11*(*5*), 583-597.

Eriksen, C. W. & Eriksen B. A. (1972). Visual backward masking as measured by voice reaction time. *Perception and Psychophysics*, *12*(*1*), 5-8.

Frankland, B. W., & Zumbo, B. D. (2002). Quantifying bimodality Part I: An easily implemented method using SPSS. *Journal of Modern Applied Statistical Methods*, *1*, 157-166.

Hartigan, J. A. (1974). Asymptotic distributions for clustering criteria. *The Annals of Statistics*, *6*(*1*), 117-131.

Hayes, (1994). *Statistics*. NY: Holt, Rinehart and Winston.

Hoffman, J. P., & Miller, A. S. (1998). Denominational influences on socially divisive issues: Polarization or continuity? *Journal for the Scientific Study of Religion*, *37*(*3*), 528-546.

Howell, D. (1995). *Statistical Methods for Psychology*. NY: Duxbury Press.

Johnson, D. N., & Yantis, S. (1995). Allocating visual attention: Tests of a two-process model. *Journal of Experimental Psychology: Human Perception & Performance*, 21(*6*), 376-1390.

Jones, P. N., & McLachlan, G. J. (1990). Algorithm AS 254: Maximum likelihood estimation from grouped and truncated data with finite mixture models. *Applied Statistics*, *39*(*2*), 273-312.

Knoll, J. L. IV, et al. (1998). Heterogeneity of the psychoses: Is there a neurodegenerative psychosis? *Schizophrenia Bulletin*, *24*(*3*), 365-379.

Müller, D. W., & Sawitzki, G. (1991). Excess mass estimates and tests for multimodality. *Journal of the American Statistical Association*, *86*, 738-746.

Ottong, S. E., & Garver, D. L. (1997). A bimodal distribution of plasma HVA/MHPG in the psychoses. *Psychiatry Research*, *69*(*2-3*), 97-103.

Reischies, F. M., Schaub, R. T., & Schlattmann, P. (1996). Normal ageing impaired cognitive functioning and senile dementia: A mixture distribution analysis. *Psychological Medicine*, *26*(*4*), 785-790.

Roeder, K. (1990). Density estimation with confidence set exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, *85*, 617-624.

Roeder, K. (1994). A graphical technique for determining the number of components in a mixture of normals. *Journal of the American Statistical Association*, *89*, 487-495.

Sussman, H. M. (1999), A neural mapping hypothesis to explain why velar stops have an allophonic split. *Brain and Language*, *70*(*2*), 294-304.

Volbrecht, V. J., Nerger, J. L., & Harlow, C. E. (1997). The bimodality of unique green revisited. *Vision Research*, *37*(*4*), 407-416.

Yantis, S., Meyer, D. E. & Smith, J. E. K. (1991). Analysis of multimodal mixture distributions: New tests for stochastic models of cognition and action. *Psychological Bulletin*, *110*, 350-374.

Yellott, J. I. (1971) Correction for fast guessing and the speed-accuracy tradeoff in choice reaction time. *Journal of Mathematical Psychology*, *8*(*2*), 159-199.